REMARKS

Claims 1-16 are now pending in the present application. Claims 1-13 are amended. Claims 14-16 are added. Reconsideration of the claims is respectfully requested.

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

I. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 1, 2, 4-9 and 12-13 under 35 U.S.C. § 103(a) as being unpatentable over *Hoyle* (US Patent No. 6,141,010) in view of *Nguyen et al.* (US Patent No. 6,202,070). This rejection is respectfully traversed.

As to claims 1, 2, 4-9 and 12-13, the Office Action states:

> **As per claim 1**, Hoyle discloses a method for maintaining software products implemented in a plurality of files in client computer systems (e.g. Fig. 3) located decentralized relative to at least one central software (e.g. *ADM Server, Ad servers* – Fig. 3) maintenance institution via a network they are connected with, such method comprising: providing product information in the network system for making it available for said client systems (e.g. *banner advertising* -- col. 8, lines 36-46; *demographic information* – col. 8, lines 53-60; *updated blueprint* – col. 13, lines 48-63); the method characterized by : performing a software maintenance action from the client site by downloading the data required for said maintenance from a set of repositories (e.g. *database 44, Ad Servers 50* – Fig. 3; col. 8, lines 47-52; col. 16, lines 37-52 – Note: accessing more than one ad servers to retrieve ad banners is equivalent to more than one repositories of banners; Fig. 13; col. 14, lines 17-26).
>
> But Hoyle does not specify that downloading is from a sequence of repositories, at least one dedicated for one particular client system, and at least one less dedicated for said client. Hoyle, however, discloses avoiding duplication and unique identifiers (e.g. col. 20, lines 47-66; col. 5, lines 26-34) and customizing according to the client local setting (e.g. col. 8, lines 55-63; col. 16, lines 9-23). Nguyen, in a method to distribute software to customize a bill of material for target machines with database management to eliminate duplicate analogous to the teaching as shown above by Hoyle's download/upgrade method, discloses the use of master database and local database at the distribution/testing site in sequence prior to software storing in the target machines and DBMs transaction operable on unique identifiers (e.g. Fig. 1; col. 4, line 65 to col. 5, line 8; col. 5, line 39 to col. 6; *software engineering group, local server database,*

> *isolated database* --line 37; col. 7, lines 22-60). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the set of repositories as mentioned by Hoyle into a sequence of repositories (DBMs) in a global and local hierarchy basis as suggested by Nguyen, because this would use enforce duplicate exclusion from operating on higher level or global software list and more local or machine specific list so as to eliminate unnecessary or redundant item; and also would prevent overhead for reconstructing of installable software list or configuration list at a more specify level of distribution hierarchy, as suggested by Nguyen.

Office Action, dated June 19, 2003. Applicant respectfully disagrees. *Hoyle* teaches a method and apparatus for providing an automatically upgradeable software application that displays targeted advertising based upon demographics and user interaction with the computer. *Hoyle* teaches a single repository of files, updated components 48, for a software product. See *Hoyle*, col. 8, line 64, to col. 9, line 11. As admitted in the Office Action, *Hoyle* fails to teach or suggest a series of repositories including at least one repository dedicated for a given client system.

Nguyen teaches a system for software distribution in computer manufacturing which manages and distributes software from release by a software engineering group to installation at a remote manufacturing site or testing facility. *Nguyen* teaches that each software engineering group's database is merged into a single master database. The master database is then replicated to form duplicate master databases at various locations. Also, software is distributed from these databases to local databases at the computer manufacturing sites and test facilities. The local databases will then be used for actual installation of the software onto personal computers. See *Nguyen*, col. 4, line 65, to col. 5, line 9. In other words, *Nguyen* teaches a distributed database management system in which master databases are replicated and ultimately distributed to local databases. However, the actual installation action comprises downloading software components **only** from the local database of the manufacturing site.

In contradistinction, the present invention provides a software maintenance action comprising downloading data required for a software maintenance action from a sequence of repositories. According to the claimed invention, the sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product. The local-level repository provides files that are specific for a given client system. Claim 1, as amended, recites:

> 1. A method for maintaining software products implemented in a plurality of files in client computer systems located decentralized relative to at least one central software maintenance institution wherein the client computer systems are connectable with the at least one central software maintenance institution via a network, the method comprising the steps of:
>
> providing product information for a product in the network system for making the product information available for said plurality of client systems; and
>
> performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system.

Neither *Hoyle* nor *Nguyen* teaches or suggests, "performing a software maintenance action for the product from a client site by downloading data required for said software maintenance action from a sequence of repositories, wherein said sequence of repositories includes at least a top-level repository storing a set of files for the product and a local-level repository storing a first subset of files for the product, wherein the first subset of files is specific for a given client system," as recited in claim 1. Since the references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claim 1 cannot be rendered obvious by the proposed combination of *Hoyle* and *Nguyen*.

Independent claims 7 and 12 recite subject matter addressed above with respect to claim 1 and are allowable for at least the same reasons addressed above. Since claims 2, 4-6, 8, 9, 13, and new claims 14-16 depend from claims 1, 7, and 12, the same distinctions between *Hoyle* and *Nguyen* and the invention recited in claims 1, 7, and 12 apply for these claims. Additionally, claims 2, 4-6, 8, 9, and 13-16 recite other additional combinations of features not suggested by the references. Consequently, it is respectfully urged that the rejection of claims 1, 2, 4-9 and 12-13 is overcome.

More particularly, claim 2 recites, "wherein the sequence of repositories includes a mid-level repository storing a second subset of files for the product, wherein the second subset of files includes at least one of a version update, a fix, and nation-specific files." Neither *Hoyle* nor *Nguyen* teaches or suggests the limitations recited in claim 2. Since the references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claim 2 cannot be rendered obvious by the proposed combination of *Hoyle* and *Nguyen*. Claims 9 and 13 recite subject matter addressed above with respect to claim 2 and are allowable for at least the same reasons.

With respect to claim 4, the Office Action states:

> **As per claim 4,** Hoyle discloses the step of upgrading with generating of an input list downloadable from a server repository (e.g. *updated blueprint* – col. 13, lines 48-63; step *256* – Fig. 14); generating a list of files present on the target client system and comparing of those lists (e.g. *current blue print* – col. 20, lines 19-32); and downloading only files which are not yet present in the target system (e.g. col. 20, lines 26-42).
>
> But Hoyle fails to specify that the downloadable input list is retrieved from at least two repositories. But in view of the combined teachings by Hoyle and Nguyen in addressing the use of a sequence of databases to improve the duplication elimination and overhead resource imparting as set forth in claim 1, this limitation herein would have been obvious for the same rationale as set forth therein.

Office Action, dated June 19, 2003. Applicant respectfully disagrees. As characterized in the Office Action, *Hoyle* only teaches a single "blue print" for downloading files. *Nguyen* actually teaches away from the present invention, because *Nguyen* teaches an elimination of duplicates. A cited portion of *Nguyen* states:

> One of the features disclosed in this software distribution system is the use of a series of databases to effect the elimination of duplicates. The system architecture is structured in a manner that eliminates duplicate software releases by the various engineering groups. Each transfer of software from database ingest to distribution at the manufacturing installation site, is utilized as an opportunity to discover and eliminate duplicate software releases.

Nguyen, col. 6, lines 10-18. Thus, it is a goal of *Nguyen* for there to be only one release of any software product in the local database of the manufacturer.

In contradistinction, the present invention encourages multiple releases, versions, fixes, and customizations to be present in the various levels of the repository hierarchy. The present invention recited in claim 4 allows a client system to select the appropriate files from each level of the repository hierarchy to achieve a proper software version that is specific to the client system. The references, taken alone or in combination, fail to teach or fairly suggest these limitations; therefore, claim 4 cannot be rendered obvious by a combination of *Hoyle* and *Nguyen*. New claim 14 recites subject matter addressed above with respect to claim 4 and is allowable for the same reasons.

Further, with respect to claim 6, the Office Action states:

> **As per claim 6,** Hoyle does not explicitly disclose a look-aside procedure to access in a neighbor system making it easier for integrating the files in the target system but discloses the local pre-store of component files (*storage 30* – Fig. 2; col. 14, lines 59-65) to alleviate unnecessary downloading of files from remote repositories. The look-aside procedure is implied by Hoyle because the technique of storing in the non-remote environment ready files for use in integrating files into the target system is thus equivalent to the technique as to look-aside for the nearest system which would facilitate the retrieval of files as intended for the upgrade because look aside is analogous to not looking further in the remote repositories.

Office Action, dated June 19, 2003. Applicant respectfully disagrees. *Hoyle* only teaches storing files **locally** on the target system for later use. The claimed feature of the look-aside procedure is not analogous to retrieving files locally. To the contrary, claim 6 expressly recites, "integrating files into the target system which have been identified by a look-aside procedure as **residing in a neighbor system.**" This limitation is neither taught nor implied in *Hoyle*. *Nguyen* fails to provide for the deficiencies of *Hoyle*. Since the references, taken alone or in combination, fail to teach or suggest each and every claim limitation, claim 6 cannot be rendered obvious by a combination of *Hoyle* and *Nguyen*. New claim 16 recites subject matter addressed above with respect to claim 6 and is allowable for the same reasons.

Therefore, the rejection of claims 1, 2, 4-9 and 12-13, under 35 U.S.C. § 103 is overcome.

Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Hoyle*, as applied to claim 1, and further in view of *Okanoue* (US Patent No. 5,689,640). Applicant notes that claim 1 is not rejected as being unpatentable over *Hoyle* alone. Therefore, a rejection of claim 3 as being unpatentable over *Hoyle* in view of *Okanoue* is improper and should be withdrawn.

As to claim 3, the Office Action states:

> **As per claim** 3, Hoyle does not disclose a fall back to an older program version by inactivate the newer version and activating the older version but teaches download and activation of downloaded components into the application (e.g. col. 14, lines 17-27). The upgrade of a software component followed by it activation and determination as to whether such activation is successful is a well-known concept in software upgrade, as evidence by Okanoue, who discloses, in a network service to update files to a plurality of target nodes, a backup copy of the original file reverted to being active if the downloaded update file fails of to activate successfully (col. 1, line 55 to col. 2, line 4; *cutover/rollback* -- Fig. 8). It would have been obvious for one of ordinary skill in the art at the time the invention was made to include the rollback step as suggested by Okanoue to the activation process by Hoyle to use the downloaded files because this would immediately and easily restore the failing system, should it encounters problems in activating the upgrade software file, to its functional state without extraneous clean-up operations or costly operating system complications by reactivating the original backup copy with its inherent machine state.

Office Action, dated June 19, 2003. Applicant respectfully disagrees. Claim 3 is allowable at least by virtue of its dependency on claim 2. *Okanoue* fails to make up for the deficiencies of *Hoyle* and *Nguyen*. Therefore, the prior art as a whole fails to teach or suggest each and every claim limitation. As such, claim 3 cannot be rendered obvious by a combination of *Hoyle* and *Okanoue*, even as combined with *Nguyen*. Therefore, the rejection of claim 3 under 35 U.S.C. § 103 is overcome.

Claims 10-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Hoyle* and *Nguyen*, as applied to claim 9, and further in view of *Nixon et al.* (US Publication No. 2003/0004952). Claim 10 is amended to be consistent with claims 6 and 16; therefore, claim 10 is allowable for the same reasons. Claim 11 is allowable at least by virtue of its dependency on claim 7. *Nixon* fails to make up for the deficiencies of *Hoyle* and *Nguyen*. Therefore, the prior art as a whole fails to teach or suggest each and every claim limitation. As such, claim 11 cannot be rendered obvious by a combination of *Hoyle*, *Okanoue*, and *Nixon*. Therefore, the rejection of claims 10 and 11 under 35 U.S.C. § 103 is overcome.

II. Conclusion

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: Sept. 19, 2003

Respectfully submitted,

Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Agent for Applicants